In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

# Suppress all warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
Data = pd.read_excel(r"C:\Users\NAGAN\Downloads\HR-Employee-Attrition.xlsx")

Data.shape #shape of the Data (rows & column)
```

Out[2]:

```
(1478, 38)
```

In [3]:

```
Data.describe()
```

Out[3]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	Em
count	1478.000000	1478.000000	1474.000000	1478.000000	1478.0	
mean	36.928958	801.702977	9.190638	2.913396	1.0	
std	9.135093	403.317966	8.093540	1.021408	0.0	
min	18.000000	102.000000	1.000000	1.000000	1.0	
25%	30.000000	465.000000	2.000000	2.000000	1.0	
50%	36.000000	801.500000	7.000000	3.000000	1.0	
75%	43.000000	1157.000000	14.000000	4.000000	1.0	
max	60.000000	1499.000000	29.000000	5.000000	1.0	

8 rows × 26 columns

In [4]:

```
Data.head() #top 5 rows
```

Out[4]:

	ID	Name	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFr
0	1Ben	Ben	41	Yes	Travel_Rarely@	1102	Sales	
1	2Nick	Nick	49	No	Travel_Frequently#	279	Research & Development	
2	3John	John	37	Yes	Travel_Rarely$	1373	Research & Development	
3	4Rock	Rock	33	No	Travel_Frequently%	1392	Research & Development	
4	5Sam	Sam	27	No	Travel_Rarely^	591	Research & Development	

5 rows × 38 columns

In [5]:

```
Data.tail() #bottom 5 rows
```

Out[5]:

	ID	Name	Age	Attrition	BusinessTravel	DailyRate	Department	Dist
1473	1469Nick	Nick	49	No	Travel_Frequently#	1023	Sales	
1474	1470John	John	34	No	Travel_Rarely#	628	Research & Development	
1475	1471Rock	Rock	27	No	Travel_Rarely@	155	Research & Development	
1476	1472Sam	Sam	49	No	Travel_Frequently#	1023	Sales	
1477	1473Jeff	Jeff	34	No	Travel_Rarely#	628	Research & Development	

5 rows × 38 columns

In [6]:

```
Data.duplicated() #check for duplicates
```

Out[6]:

```
0        False
1        False
2        False
3        False
4        False
         ...
1473      True
1474      True
1475      True
1476      True
1477      True
Length: 1478, dtype: bool
```

In [7]:

```
df = Data.drop_duplicates() # remove duplicates
df
```

Out[7]:

	ID	Name	Age	Attrition	BusinessTravel	DailyRate	Department	Dist
0	1Ben	Ben	41	Yes	Travel_Rarely@	1102	Sales	
1	2Nick	Nick	49	No	Travel_Frequently#	279	Research & Development	
2	3John	John	37	Yes	Travel_Rarely$	1373	Research & Development	
3	4Rock	Rock	33	No	Travel_Frequently%	1392	Research & Development	
4	5Sam	Sam	27	No	Travel_Rarely^	591	Research & Development	
...	...	...	...	...	...	...	...	
1468	1469Nick	Nick	49	No	Travel_Frequently#	1023	Sales	
1469	1470John	John	34	No	Travel_Rarely#	628	Research & Development	
1470	1471Rock	Rock	27	No	Travel_Rarely@	155	Research & Development	
1471	1472Sam	Sam	49	No	Travel_Frequently#	1023	Sales	
1472	1473Jeff	Jeff	34	No	Travel_Rarely#	628	Research & Development	

1473 rows × 38 columns

In [8]:

```
df = df.drop(columns=['ID', 'Name']) # remove columns
df.head(5)
```

Out[8]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Educ
0	41	Yes	Travel_Rarely@	1102	Sales	1.0	
1	49	No	Travel_Frequently#	279	Research & Development	8.0	
2	37	Yes	Travel_Rarely$	1373	Research & Development	2.0	
3	33	No	Travel_Frequently%	1392	Research & Development	3.0	
4	27	No	Travel_Rarely^	591	Research & Development	2.0	

5 rows × 36 columns

In [9]:

```
# Remove unwanted characters from the specified column using .loc
df.loc[:, 'BusinessTravel'] = (df['BusinessTravel'].str.replace(r'[_\W]+', " ",

df.head(5)
```

Out[9]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Educatic
0	41	Yes	Travel Rarely	1102	Sales	1.0	
1	49	No	Travel Frequently	279	Research & Development	8.0	
2	37	Yes	Travel Rarely	1373	Research & Development	2.0	
3	33	No	Travel Frequently	1392	Research & Development	3.0	
4	27	No	Travel Rarely	591	Research & Development	2.0	

5 rows × 36 columns

In [10]:

```
df["Joining_date"].head(5) # check for the date format
df["Joining_date"]
```

Out[10]:

```
0       26/07/2018 00:00:00
1       08/09/2020 00:00:00
2       07/09/2014 00:00:00
3       09/08/2018 00:00:00
4       13/09/2021 00:00:00
                ...
1468    26/08/2015 00:00:00
1469    07/08/2020 00:00:00
1470    02/08/2018 00:00:00
1471    26/08/2015 00:00:00
1472    07/08/2020 00:00:00
Name: Joining_date, Length: 1473, dtype: object
```

In [11]:

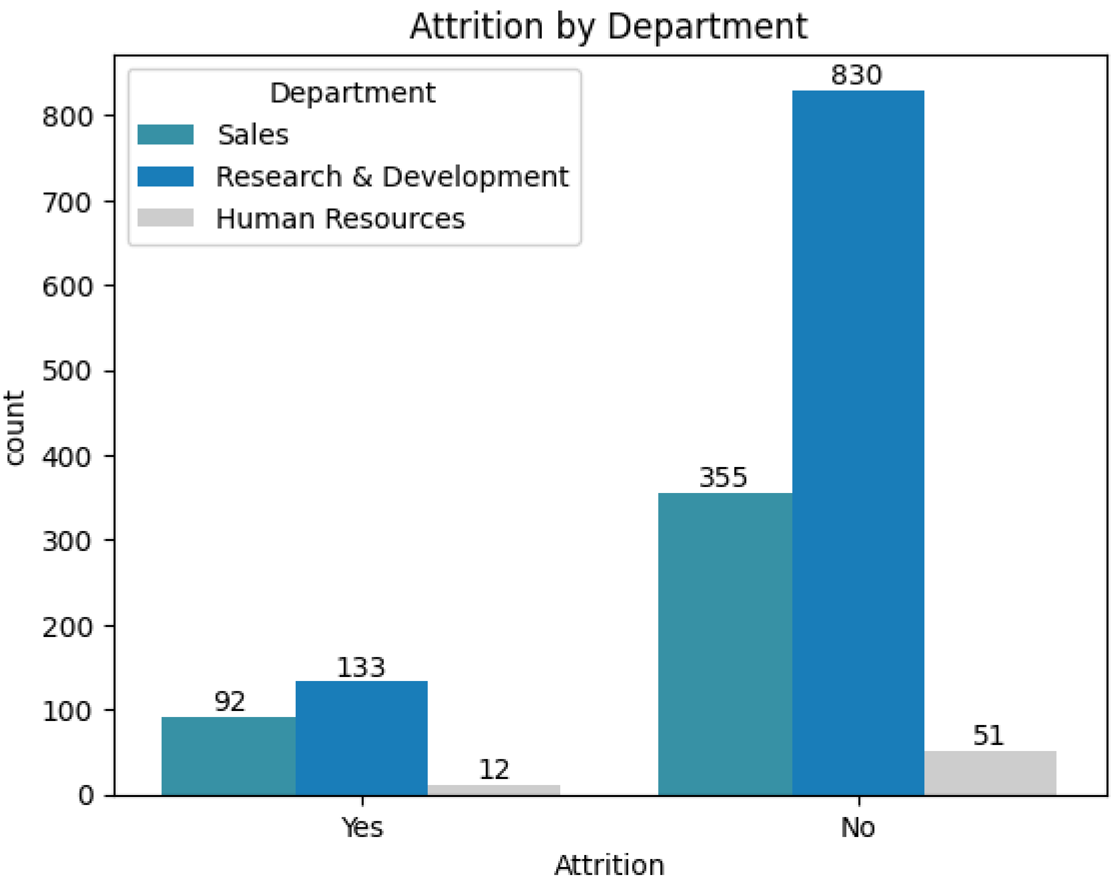
```
#remove whitespace & convert to date format

df["Joining_date"] = df["Joining_date"].astype(str).str.strip()

df["Joining_date"] = pd.to_datetime(df["Joining_date"])

df["Joining_date"].head(5)
```

Out[11]:

```
0   2018-07-26
1   2020-09-08
2   2014-09-07
3   2018-08-09
4   2021-09-13
Name: Joining_date, dtype: datetime64[ns]
```

In [12]:

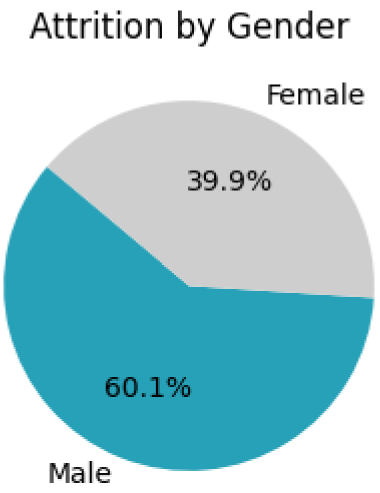
```
palette = {
    'Sales': '#27A1B7',
    'Human Resources': '#CECECE',
    'Research & Development':'#0187D4'
}
%matplotlib inline
sns.countplot(x='Attrition', hue='Department', data=df,palette=palette)
plt.title('Attrition by Department')
for container in plt.gca().containers:
    plt.gca().bar_label(container)
plt.show()
```



In [13]:
```
gender_counts = df['Gender'].value_counts() # gender count
gender_counts
```

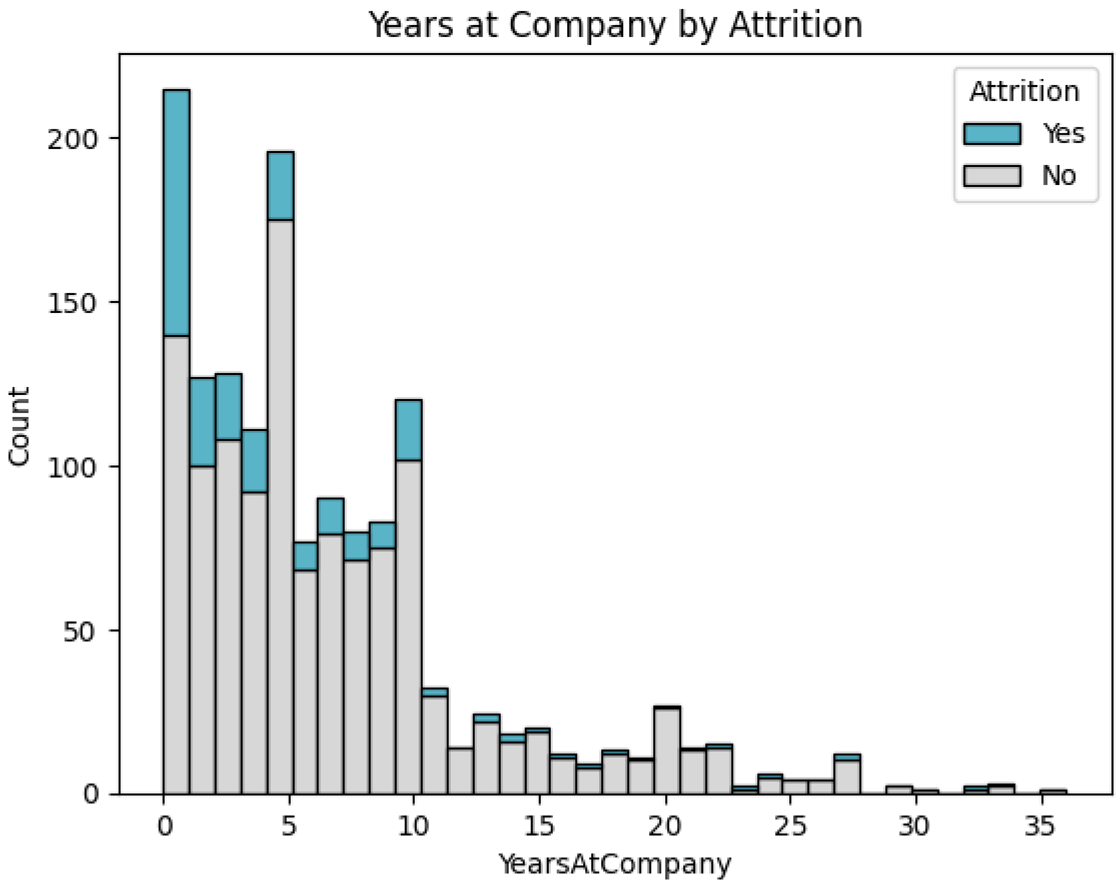
Out[13]:
```
Gender
Male      885
Female    588
Name: count, dtype: int64
```

In [14]:
```
colors = ['#27A1B7','#CECECE']

plt.figure(figsize=(3, 3))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle
plt.title('Attrition by Gender')
plt.show()
```



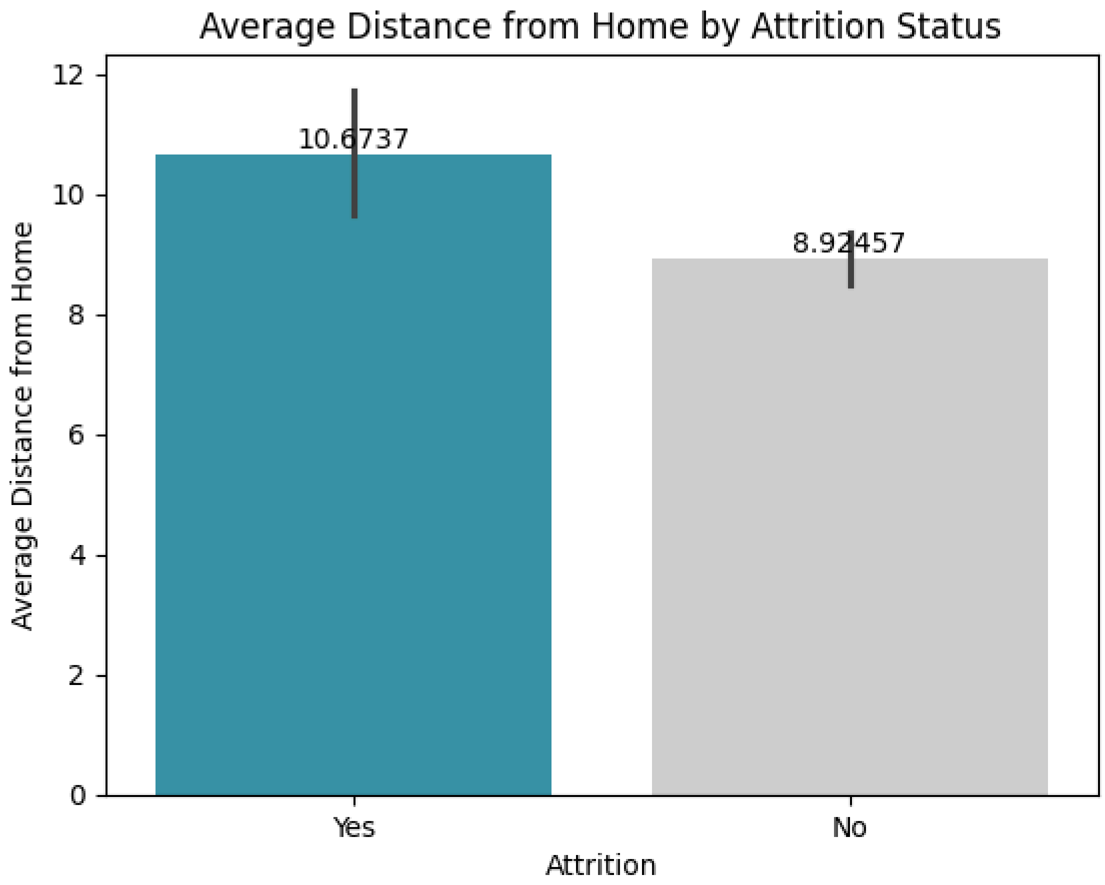
In [15]:

```
palette = {'Yes': '#27A1B7', 'No': '#CECECE'}

sns.histplot(data=df, x='YearsAtCompany', hue='Attrition', multiple='stack',pale
plt.title('Years at Company by Attrition')
plt.show()
```



In [16]:

```
palette = {'Yes': '#27A1B7', 'No': '#CECECE'}

ax = sns.barplot(data=df, x='Attrition', y='DistanceFromHome', estimator='mean',
for container in ax.containers:
    ax.bar_label(container)
plt.title('Average Distance from Home by Attrition Status')
plt.ylabel('Average Distance from Home')
plt.xlabel('Attrition')
plt.show()
```

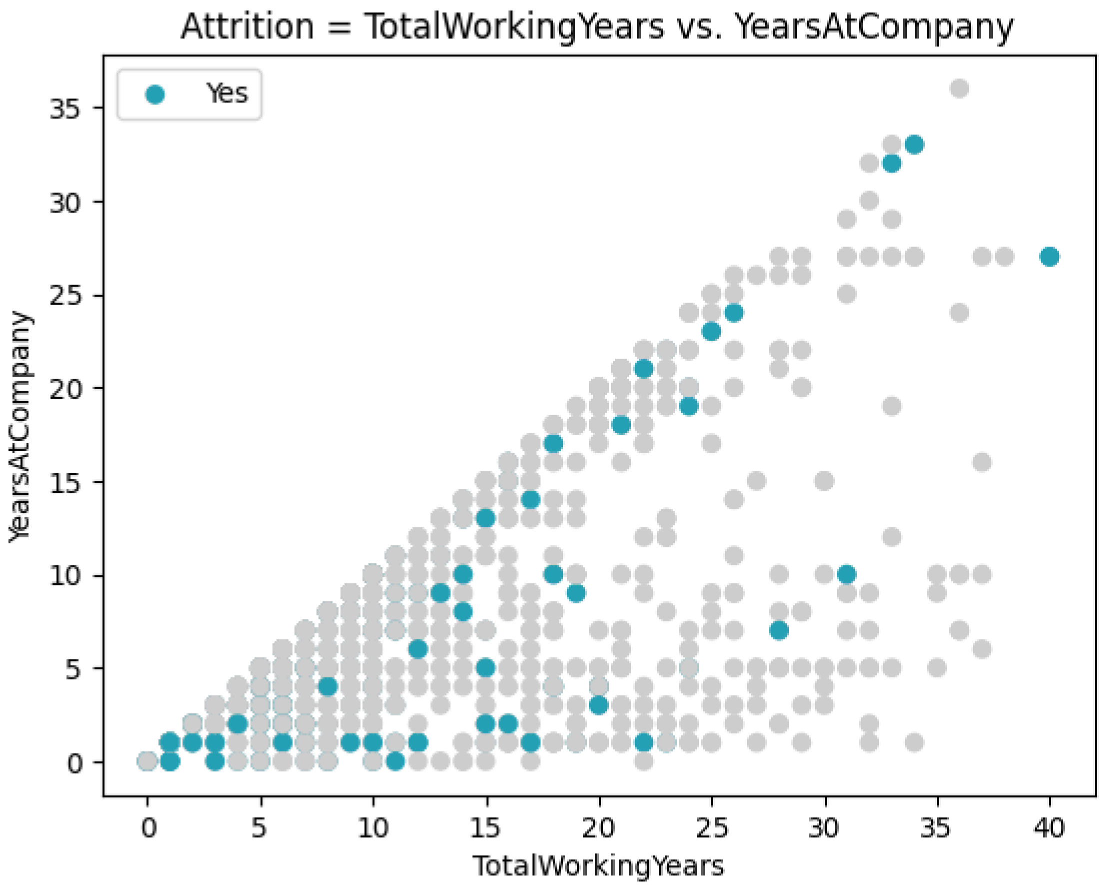


In [17]:

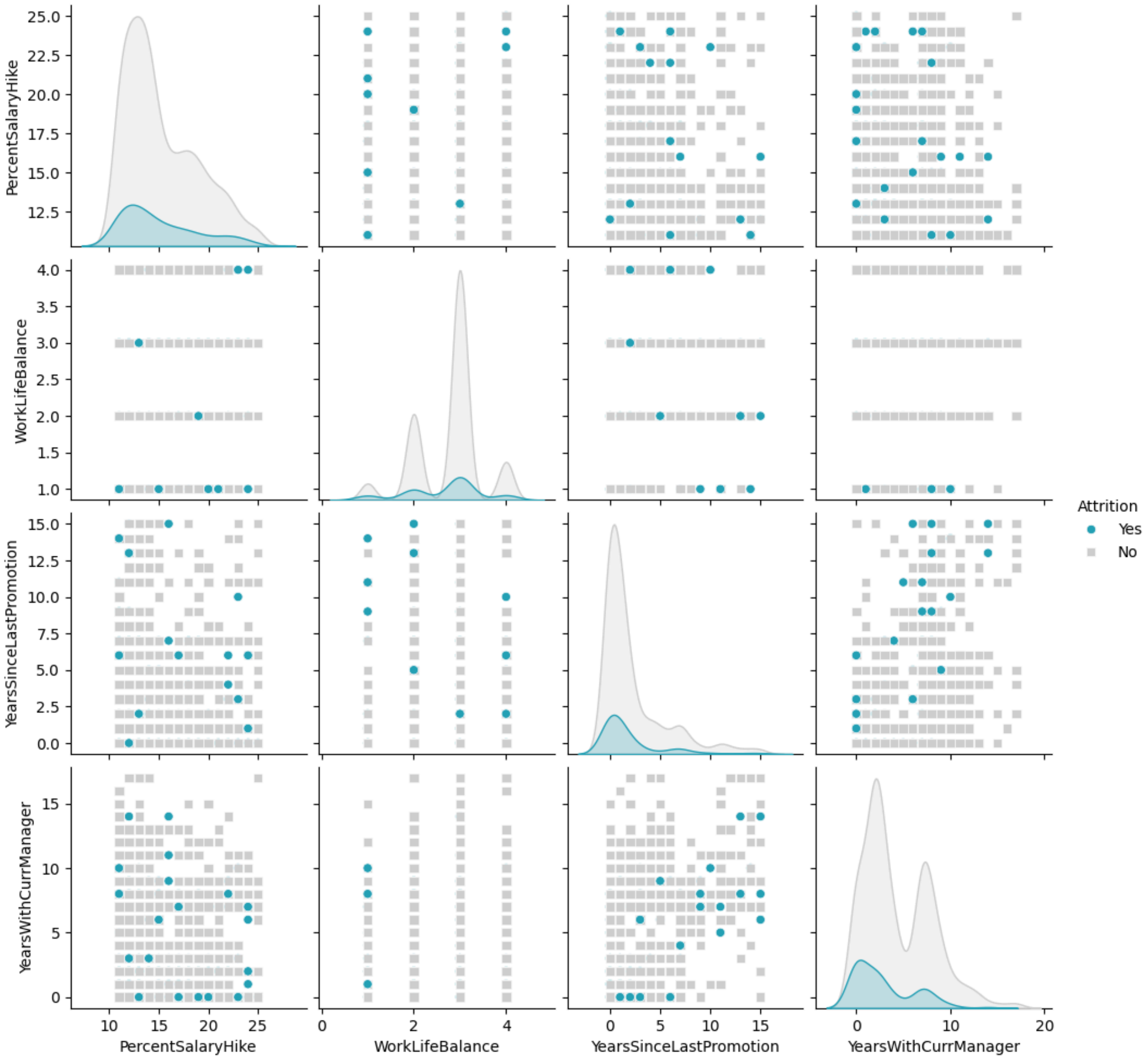
```
color_map = {'Yes': '#27A1B7', 'No': '#CECECE'}
scatter_colors = df['Attrition'].map(color_map)

x = df['TotalWorkingYears']
y = df['YearsAtCompany']

plt.scatter(x, y, c=scatter_colors)
plt.xlabel('TotalWorkingYears')
plt.ylabel('YearsAtCompany')
plt.legend(df['Attrition'])
plt.title('Attrition = TotalWorkingYears vs. YearsAtCompany')
```

Out[17]:

```
Text(0.5, 1.0, 'Attrition = TotalWorkingYears vs. YearsAtCompany')
```



```
In [24]: # List of selected columns
selected_columns = [
    "PercentSalaryHike",
    "WorkLifeBalance", "YearsSinceLastPromotion", "YearsWithCurrManager",
    "BusinessTravel", "Attrition" ] # Assuming 'Attrition' is for the hue


# Filter the DataFrame to include only the selected columns
df_filtered = df[selected_columns]

# Plotting
sns.pairplot(df_filtered, hue='Attrition', palette={'Yes': '#27A1B7', 'No': '#CE
plt.show()
```


PercentSalaryHike
25.0
22.5
20.0
17.5
15.0
12.5
WorkLifeBalance
4.0
3.5
3.0
2.5
2.0
1.5
1.0
YearsSinceLastPromotion
15.0
12.5
10.0
7.5
5.0
2.5
0.0
YearsWithCurrManager
15
10
5
0
10 15 20 25
PercentSalaryHike
0 2 4
WorkLifeBalance
0 5 10 15
YearsSinceLastPromotion
0 10 20
YearsWithCurrManager
Attrition
● Yes
■ No